

Efficient mass and stiffness matrix assembly via weighted Gaussian quadrature rules for B-splines

Michael Barton^{a,*}, Vladimir Puzyrev^{b,c}, Quanling Deng^{b,c}, Victor Calo^{b,d,c}

^a*BCAM – Basque Center for Applied Mathematics, Alameda de Mazarredo 14,
48009 Bilbao, Basque Country, Spain*

^b*Department of Applied Geology, Faculty of Science and Engineering, Curtin University,
Kent Street, Bentley, Perth, WA 6102, Australia*

^c*Curtin Institute for Computation, Curtin University, Kent Street,
Bentley, Perth, WA 6102, Australia*

^d*Mineral Resources, Commonwealth Scientific and Industrial Research Organization
(CSIRO), Kensington, Perth, WA 6152, Australia*

Abstract

Calabrò et al. [10] changed the paradigm of the mass and stiffness computation from the traditional element-wise assembly to a row-wise concept, showing that the latter one offers integration that may be orders of magnitude faster. Considering a B-spline basis function as a non-negative measure, each mass matrix row is integrated by its own quadrature rule with respect to that measure. Each rule is easy to compute as it leads to a linear system of equations, however, the quadrature rules are of the Newton-Cotes type, that is, they require a number of quadrature points that is equal to the dimension of the spline space. In this work, we propose weighted quadrature rules of Gaussian type which require the minimum number of quadrature points while guaranteeing exactness of integration with respect to the weight function. The weighted Gaussian rules arise as solutions of non-linear systems of equations. We derive rules for the mass and stiffness matrices for uniform C^1 quadratic and C^2 cubic isogeometric discretizations. In each parameter direction, our rules require locally only $p + 1$ quadrature points, p being the polynomial degree. While the nodes cannot be reused for various weight functions as in [10], the computational cost of the mass and stiffness matrix assembly is comparable.

Keywords: weighted Gaussian quadrature, B-splines, isogeometric analysis, mass and stiffness matrix assembly

*Corresponding author

Email addresses: mbarton@bcamath.org (Michael Barton),
Vladimir.Puzyrev@Curtin.edu.au (Vladimir Puzyrev), Quanling.Deng@Curtin.edu.au
(Quanling Deng), Victor.Calo@Curtin.edu.au (Victor Calo)

1. Introduction

Systems of Partial Differential Equations (PDEs) describe many relevant physical processes. These physical phenomena are traditionally modeled using finite element analysis (FEA) and Isogeometric analysis (IGA) [12]. Numerical integration is a fundamental step in the assembly process of the algebraic systems that result from the FEA and IGA discretizations. Depending on the type of the governing PDE, the matrix assembly requires integration of products of basis functions (mass matrix) and/or their gradients (stiffness matrix) which, in the context of IGA, requires integration of spline spaces of a certain structure [7]. Efficient quadrature rules play a key role in this process, as they are cheap and elegant tools to exactly integrate the space of functions under consideration [9, 16, 19, 25].

Traditionally in the tensor product based IGA, mass and stiffness matrix assembly is performed element-wise, using a corresponding univariate quadrature rule on each element in each parametric direction. While using standard polynomial Gauss quadrature on each element is a usual approach in many FEA and IGA codes, Gaussian quadrature for splines offer a lot cheaper alternative as the higher continuity between elements signifies that fewer Gaussian quadrature points are needed [24, 28]. For example, for the C^1 quadratic spline space, the mass matrix contains terms that belong to a quartic C^1 space. Gaussian quadrature for this space requires asymptotically, i.e. for a large number of elements, only one and a half quadrature points per element in contrast to the polynomial Gauss rule that requires three quadrature points per element [7].

In the IGA community, several recent papers focused on the development of efficient quadrature rules [1, 3, 9, 19]. Hughes et al. [19] introduced efficient rules that are exact over the whole real line (infinite domain). For finite domains, one may introduce additional quadrature points [3] which make the rule non-Gaussian (slightly sub-optimal in terms of quadrature points), but more importantly, it yields quadrature weights that can be negative, unlike Gaussian quadratures.

To overcome these drawbacks, alternative quadrature schemes were introduced in recent years [2, 5, 7, 8, 18, 20, 27]. In these studies, the sought Gaussian quadrature rule is represented as a root of a piece-wise polynomial system, where the system expresses the exactness of the rule when applied to a basis of the underlying spline space. In general, the system is highly non-linear and computing a root numerically using, e.g., Newton-Raphson may not always converge [20], unless a very good initial guess is known. Finding a good initial guess, however, is possible only for specific target spaces using the local structure of the B-spline basis [1, 18].

An alternative good initial guess has been proposed by using the continuity argument between a spline space and its Gaussian quadrature [6]. A Gaussian quadrature rule for the desired spline space is derived from a known Gaussian quadrature (e.g., a union of polynomial Gauss rules) by continuously modifying the source knot vector into the target one. The process is effective for arbitrary knot vectors (including non-uniform spacing and arbitrary continuity) and poly-

nomial degrees. The sought quadrature rule corresponds to a zero of a piece-wise polynomial system and is traced numerically as the knot vector changes from the source to the target one. This homotopic continuation approach is used for spline spaces of various degrees and continuities [5, 7], showing also the numerical evidence that, for uniform knot vectors, the rules over finite domains quickly converge to the half-point rules of Hughes et al. over infinite domains [19].

A quadrature-free approach to assemble mass and stiffness matrices uses the observation that exact integration is not required to achieve the optimal convergence rate of the solution. Therefore, the integrals arising from the geometry factor can be approximated by the integrals of the B-spline basis functions, which are precomputed and stored in a look-up table [22, 23]. Another efficient alternative is the variational collocation method [17]. In [17], the authors prove the existence of Cauchy-Galerkin collocation points, that is, points in which the collocated solution reproduces the Galerkin counterpart. Therefore, such a method has a great advantage as it possesses the exactness of Galerkin solution for the cost of collocation. However, a stable computational framework to efficiently determine the Cauchy-Galerkin points is an open challenge, particularly for spline spaces of various continuities and non-uniform knot vectors [15].

Recently, Calabrò et al. [10] have changed the paradigm of the mass and stiffness computation from the traditional element-wise assembly to a row-wise concept. When building the mass matrix, one B-spline basis function of the scalar product is considered as a positive measure (i.e., a weight function), and a weighted quadrature with respect to that weight is computed for each matrix row. Such an approach brings significant computational savings compared to the traditional approaches that use Gaussian or semi-Gaussian element-wise assembly [5, 7, 18, 20, 27], because in these integration schemes, the number of quadrature points contains a p^d term, p being the polynomial degree and d the dimension. In contrast, [10] requires in each parametric direction in the limit only two points per element, regardless the polynomial degree. For each weight (mass matrix row), its specific weighted quadrature is computed by solving a linear system. These rules, however, are quadratures of the Newton-Cotes type, that is, they require the same number of quadrature points as the spline basis functions involved.

In this work, we propose weighted quadrature rules of Gaussian type which require the minimum number of quadrature points while guaranteeing the exactness of integration with respect to the weight function. Locally, our rules require only $p + 1$ quadrature points and, even though our quadrature points cannot be reused like in [10], our quadratures still perform similarly in terms of floating-point operations (FLOPs) when compared to [10] due to the lower number of summands ($p + 1$ vs $2p + 1$) in the quadrature rule. Moreover, there are situations such as parallel computation of matrix assembly when each matrix entry can be sent to a different core and reuse of quadrature points is hardly possible.

We derive rules for C^1 quadratic and C^2 cubic isogeometric discretizations over uniform knot vectors. The rules arise as solutions of the resulting nonlinear systems of equations. The solution to the quadratic case can be expressed

symbolically. For cubic splines, the rules are computed numerically using the Newton-Raphson method.

The rest of the paper is organized as follows. Section 2 formulates the model problem and its isogeometric discretization. Then we derive the weighted Gaussian quadrature rules for quadratic and cubic spline spaces in Section 3 and show numerical experiments that validate our theoretical results in Section 4. We conclude the paper and indicate directions for future research in Section 5.

2. Preliminaries

We set up our model problem, derive its isogeometric discretization, and define spline spaces that we need for the approximation of the solution.

2.1. Model problem

We consider the Poisson problem

$$\begin{cases} -\Delta u = f & \text{in } \Omega, \\ u = 0 & \text{on } \partial\Omega \end{cases} \quad (1)$$

as a model problem with a physical domain $\Omega \subset \mathbb{R}^d$, $d = 1, 2, 3$, with bounded Lipschitz boundary $\partial\Omega$. Let $V = H_0^1(\Omega)$, the variational form of (1) is to find $u \in V$ such that

$$a(u, v) = l(v) \quad \forall v \in V, \quad (2)$$

where the bilinear and linear forms are

$$a(u, v) = \int_{\Omega} \nabla u^T(\mathbf{x}) \cdot \nabla v(\mathbf{x}) \, d\mathbf{x} \quad \text{and} \quad l(v) = \int_{\Omega} f(\mathbf{x})v(\mathbf{x}) \, d\mathbf{x}, \quad (3)$$

respectively.

2.2. Isogeometric discretization

In the isogeometric framework, Ω is described with the same basis as the solution space, that is, it is an isoparametric discretization. In this case, we commonly use B-spline or NURBS parameterizations. Consider a single patch geometry map $G : \hat{\Omega} \rightarrow \Omega$, where the parameter domain $\hat{\Omega}$ is a unit box in \mathbb{R}^d , $\hat{\Omega} = [0, 1]^d$. The mapping G maps any point $\hat{\mathbf{x}} \in \hat{\Omega}$ from the parameter domain to the physical domain via

$$\mathbf{x} = G(\hat{\mathbf{x}}) = \sum_{\mathbf{i} \in \mathcal{I}} \mathbf{g}_{\mathbf{i}} \hat{B}_{\mathbf{i}}(\hat{\mathbf{x}}), \quad (4)$$

where $\mathbf{g}_{\mathbf{i}}$ are the spline (NURBS) control points, $\hat{B}_{\mathbf{i}}$ are the basis functions, and \mathbf{i} is a d -dimensional multi-index, i.e., $\mathbf{i} = (i_1, \dots, i_d)$.

Remark 1. For simplicity, we consider only a single patch geometry map G in this work. We refer the reader to [21, 29] for multipatch parametrization techniques.

The basis functions possess the tensor product structure and we write

$$\hat{B}_{\mathbf{i}}(\hat{\mathbf{x}}) = \hat{B}_{i_1}(\hat{x}_1) \dots \hat{B}_{i_d}(\hat{x}_d), \quad (5)$$

that are assumed, for the simplicity of the argument, to be piece-wise polynomial functions of the same degree p in every variable. Let us denote by N_i^{EL} the number of elements in the i -th parametric direction and define

$$\Xi_i = (0 = \underbrace{\xi_0, \dots, \xi_0}_{p+1}, \xi_1, \xi_2, \dots, \xi_{N_i^{\text{EL}}-1}, \underbrace{\xi_{N_i^{\text{EL}}}, \dots, \xi_{N_i^{\text{EL}}}}_{p+1} = 1) \quad (6)$$

the knot vector in the i -th direction. We assume that all the univariate splines have open knot vectors and are of the maximum continuity C^{p-1} , i.e., all the internal knots $\xi_1, \dots, \xi_{N_i^{\text{EL}}-1}$ are single knots.

Remark 2. *Conceptually, one can consider various degrees and continuities for univariate splines in each parametric direction. Assuming the same degree and continuity is not a limitation of the proposed method, it only simplifies the argument and offers a more convenient implementation.*

We denote by \hat{S}_i the spline space of degree p over a knot vector Ξ_i . For its dimension we get

$$\dim(\hat{S}_i) = p + N_i^{\text{EL}} \quad i = 1, \dots, d, \quad (7)$$

and, due to the tensor product structure, the total number of degrees of freedom

$$N^{\text{DOF}} = \prod_{i=1}^d \dim(\hat{S}_i) \quad (8)$$

is the dimension of the approximate solution space

$$V_h = \text{span}\{\phi_{\mathbf{i}}, \mathbf{i} \in \mathcal{I}\} \quad \text{with} \quad \phi_{\mathbf{i}}(\mathbf{x}) = \hat{B}_{\mathbf{i}}(G^{-1}(\mathbf{x})), \quad \mathbf{x} \in \Omega. \quad (9)$$

The Galerkin projection transforms the variational formulation (2) into finding

$$u_h \in V_h \quad \text{such that} \quad a(u_h, v_h) = l(v_h) \quad \forall v \in V_h, \quad (10)$$

where h is the maximum element size. The elements of V_h can be written as linear combinations of basis functions. Herein, for u_h , we write

$$u_h = \sum_{\mathbf{i} \in \mathcal{I}} u_{\mathbf{i}} \phi_{\mathbf{i}}, \quad (11)$$

with the coefficients vector $\mathbf{u} = (u_1, \dots, u_{N^{\text{DOF}}})$. We further introduce the pull-backs of the functions that are defined in the physical domain as

$$\hat{u}_h(\hat{\mathbf{x}}) = u_h(G(\hat{\mathbf{x}})) \quad \hat{f}(\hat{\mathbf{x}}) = f(G(\hat{\mathbf{x}})) \quad \hat{\phi}(\hat{\mathbf{x}}) = \phi(G(\hat{\mathbf{x}})), \quad \hat{\mathbf{x}} \in \hat{\Omega} \quad (12)$$

and denote by J the Jacobian matrix $J = \hat{\nabla} G$, $\hat{\nabla}$ being the gradient in the parameter domain. Using the two forms (3), then the coefficients of u_h are a solution of the linear system

$$\mathbf{K} \mathbf{u} = \mathbf{b}, \quad (13)$$

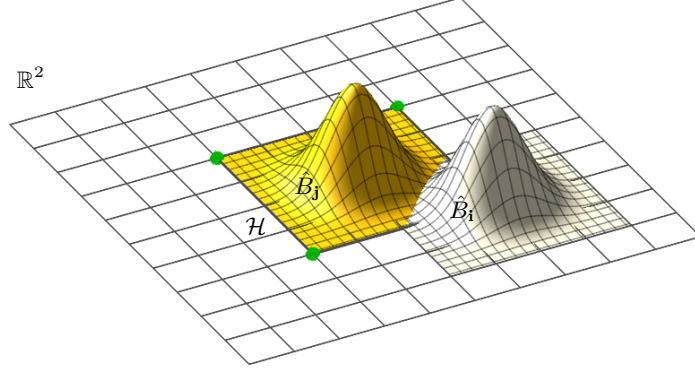


Figure 1: Spline basis functions in the neighborhood of the weight function; $p = 3$, $d = 2$. The 2D-grid defines a knot neighborhood (here uniform) of $(2p + 1)^d$ basis functions \hat{B}_i that have non-vanishing integrals with respect to the measure \hat{B}_j , see (17).

where \mathbf{K} is the *stiffness matrix*

$$\mathbf{K}_{ij} = a(\phi_i, \phi_j) = \int_{\Omega} \hat{\nabla} \hat{B}_i(\hat{\mathbf{x}})^T J(\hat{\mathbf{x}})^{-1} J(\hat{\mathbf{x}})^{-T} \hat{\nabla} \hat{B}_j(\hat{\mathbf{x}}) |\det(J(\hat{\mathbf{x}}))| d\hat{\mathbf{x}}, \quad (14)$$

and \mathbf{b} is the load vector

$$\mathbf{b}_i = l(\phi_i) = \int_{\Omega} \hat{f} \hat{B}_i(\hat{\mathbf{x}}) |\det(J(\hat{\mathbf{x}}))| d\hat{\mathbf{x}}. \quad (15)$$

For partial differential equations that contain zero-order terms, the variational form contains also scalar products of basis functions that form the *mass matrix*

$$\mathbf{M}_{ij} = b(\phi_i, \phi_j) = \int_{\Omega} \hat{B}_i(\hat{\mathbf{x}}) \hat{B}_j(\hat{\mathbf{x}}) |\det(J(\hat{\mathbf{x}}))| d\hat{\mathbf{x}}. \quad (16)$$

3. Integration via weighted quadrature

We start our considerations with the numerical integration of the type

$$\int_{\Omega} \hat{B}_i(\hat{\mathbf{x}}) \hat{B}_j(\hat{\mathbf{x}}) d\hat{\mathbf{x}}, \quad (17)$$

which, due to the tensor product structure of $\hat{B}_i(\hat{\mathbf{x}})$, can be decomposed into a sequence of d univariate integrations

$$\int_0^1 \hat{B}_{i_1}(\hat{x}_1) \hat{B}_{j_1}(\hat{x}_1) \left[\int_0^1 \hat{B}_{i_2}(\hat{x}_2) \hat{B}_{j_2}(\hat{x}_2) \cdots \left[\int_0^1 \hat{B}_{i_d}(\hat{x}_d) \hat{B}_{j_d}(\hat{x}_d) d\hat{x}_d \right] \cdots d\hat{x}_2 \right] d\hat{x}_1. \quad (18)$$

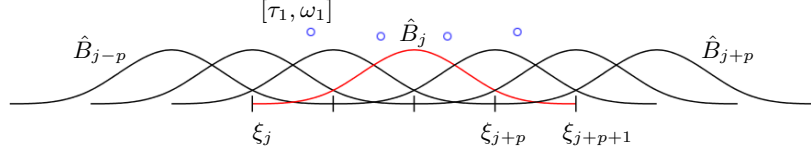


Figure 2: Weighted Gauss quadrature for univariate cardinal B-splines, $p = 3$. Gaussian quadrature with respect to a measure \hat{B}_j (red) requires $p + 1$ quadrature points (blue dots) since only $2p + 1$ basis functions $\hat{B}_{j-p}, \dots, \hat{B}_{j+p}$ have an overlapping support with the one of \hat{B}_j , $[\xi_j, \xi_{j+p+1}]$.

We follow [10] where one basis function is the weight function for the integration. Due to the local properties of B-spline basis functions, the measure is positive only on the support of \hat{B}_j and zero elsewhere. Defining $\mathcal{H} = \text{supp}(\hat{B}_j)$, $\mathcal{H} \subset \hat{\Omega}$, a weighted Gaussian quadrature with respect to the measure \hat{B}_j can be seen as a local quadrature mask acting only on the macroelement \mathcal{H} , see Fig. 1.

The decomposition (18) allows us to consider a sequence of univariate integrals. We aim to compute

$$\int_{\text{supp}(\hat{B}_j)} \hat{B}_i(\hat{x}) \hat{B}_j(\hat{x}) d\hat{x} \quad (19)$$

by deriving a Gauss quadrature rule with respect to a non-negative measure $\mu = \hat{B}_j(\hat{x}) d\hat{x}$. The Gauss must be exact for all B-spline functions \hat{B}_i that have non-zero support on $\text{supp}(\hat{B}_j)$, see Fig. 2.

The number of spline basis functions that have non-zero support on $\text{supp}(\hat{B}_j)$ is $2p + 1$ and therefore the weighted Gauss quadrature requires only $p + 1$ quadrature points. Since \hat{B}_j spans $p + 1$ elements, this results in a one-node-per-element rule, regardless the degree. Therefore integrating w.r.t one weight function gives, due to the tensor product decomposition (18), a reduction factor

$$\left(\frac{p+1}{2p+1} \right)^d. \quad (20)$$

However, the scheme [10] allows node re-use, which is a significant advantage. For repeated integration with various weights (the whole matrix assembly), the comparison of the number of function evaluations per element reads as

$$\left(\frac{p+1}{2} \right)^d \quad (21)$$

in the favor of [10].

Remark 3. The reduction in (20) compares solely the number of quadrature points between our approach and [10] when integrating one basis function with respect to one specific weight. Therefore, it states only local reduction factor. Since the quadrature points of [10] are the knots and midpoints, the evaluations of various basis functions at the quadrature points can be reused while our

quadrature points differ for every weight function. Nonetheless, Section 4 shows that globally the computational cost in terms of FLOPs is comparable due to the fact that our rules consist of $p + 1$ summands in contrast to $2p + 1$ of [10].

We now derive specific rules for uniform C^1 quadratic and C^2 cubic isogeometric discretizations. With a slight abuse of notation, we omit the hat symbol over the basis functions and variables, remembering that all belong to the parameter domain. Note that conceptually one could follow the homotopic continuation approach and compute the nodes even for non-uniform meshes, however, such an approach is computationally demanding and is not expected to bring computational gain over [10]. The advantage of the presented weighted Gaussian rules over uniform meshes is their repetitivity and the fact that they can be read from a small look-up table.

3.1. Weighted Gaussian quadrature – Mass matrix terms

We seek an m -point Gauss quadrature rule

$$\int_a^b f(x)w(x) dx = \sum_{i=1}^m \omega_i f(\tau_i) + R_m(f), \quad (22)$$

with respect to the non-negative weight function $w(x)$ such that the rule is exact for any function $f(x)$ from a spline space under consideration. The existence and uniqueness of such a rule has been investigated in [24]. In our context, the weight function is represented by the spline basis function \hat{B}_j , the domain of integration is its support $[\xi_j, \xi_{j+p+1}]$, for some $j \in \mathbb{Z}$, and the number of Gaussian points is $m = p + 1$. To meet the exactness constraint, we obtain a system

$$\int_{\xi_j}^{\xi_{j+p+1}} B_i(x)B_j(x) dx = \sum_{k=1}^{p+1} \omega_k^j B_i(\tau_k^j) \quad \forall i = j - p, \dots, j + p,^1 \quad (23)$$

where the quadrature points τ_k^j and weights ω_k^j relate to the weight function B_j and are the unknowns of an under-constrained system that consists of $2p + 1$ equations and $2(p + 1)$ unknowns.

Remark 4. According to [24, Theorem 3.1], there exists a unique quadrature rule for splines with respect to a positive measure if the corresponding system is well-constrained (same number of unknowns and constraints). Our rules are Gaussian as they guarantee exactness with a minimum number of quadrature points, however, they are not unique as the rules are roots of underconstrained piece-wise polynomial systems.

¹The indexing is for an internal weight function, i.e., a weight outside boundary. For the weight functions affected with the boundary, one has to take into account the knot multiplicity to get the correct index i .

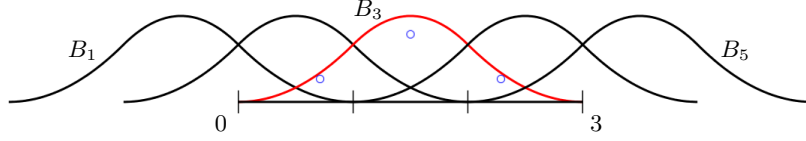


Figure 3: Weighted Gaussian quadrature for C^1 quadratic cardinal spline ($p = 2$); mass matrix term. The weight function B_3 (red) vanishes outside $[0, 3]$ and has non-trivial overlap with itself and four other basis functions. The Gaussian quadrature requires $p+1$ quadrature points (blue dots, (25)) that, due to symmetry, arise as a solution of the (3×3) system (24).

3.1.1. C^1 quadratic elements

We start with the quadratic case ($p = 2$). Consider the cardinal B-spline basis function B_j with a support on $[0, 3]$. There are five basis functions that interact with B_j and therefore one needs three quadrature points to satisfy the conditions (23), see Fig. 3. There is one degree of freedom to choose either one quadrature point or weight. We impose the symmetry constraint and set $\tau_2 = 1.5$. Consequently, we solve a reduced, well-constrained (3×3) system

$$\begin{aligned} \frac{1}{2}\omega_1(1 - \tau_1)^2 &= \frac{1}{120}, \\ \frac{1}{2}\omega_1 - \omega_1\tau_1^2 + \omega_1\tau_1 + \frac{1}{8}\omega_2 &= \frac{13}{60}, \\ \omega_1\tau_1^2 + \frac{3}{4}\omega_2 &= \frac{11}{20} \end{aligned} \tag{24}$$

with the unknowns τ_1 , ω_1 , and ω_2 . We solve (24) symbolically by using Maple. Evaluating the solutions with a precision of twenty decimal digits, we obtain the weighted Gaussian rule

$$\begin{aligned} \tau_1 &= 0.71241440095955149482, & \omega_1 &= 0.20151829499655592436, \\ \tau_2 &= 1.5, & \omega_2 &= 0.59696341000688815128. \end{aligned} \tag{25}$$

In the non-uniform setting, one loses the symmetry on the elements, thus cannot reduce the system to (3×3) as is done in (24). Instead, one sets and solves an analogous (6×6) system, having the three nodes and weights as the unknowns. The existence of a solution of such a system is guaranteed by the result of [24].

3.1.2. C^2 cubic elements

For the cubic case, we proceed analogously. The Gaussian quadrature for cubic cardinal B-splines on $[0, 4]$ with a weight B_j requires $p + 1 = 4$ nodes, see Fig. 2. Due to symmetry, the system (23) can be reduced to a well-constrained

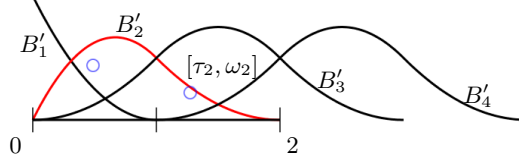


Figure 4: Weighted Gaussian quadrature for a boundary weight for C^1 quadratic ($p = 2$) cardinal spline. The boundary weight function B'_2 (red) vanishes outside $[0, 2]$ and has non-trivial overlap with itself and three other basis functions. The Gaussian quadrature nodes and weights with respect to B'_2 are the solution of the system (28).

(4×4) system

$$\begin{aligned}
\frac{1}{6}\omega_1(1 - \tau_1)^3 &= \frac{1}{5040}, \\
\frac{1}{6}\omega_2(2 - \tau_2)^3 + \frac{1}{42}\omega_1(21\tau_1^3 - 42\tau_1^2 + 28) &= \frac{1}{42}, \\
\frac{1}{1680}\omega_1(-840\tau_1^3 + 840\tau_1^2 + 840\tau_1 + 280) + & \\
\frac{1}{1680}\omega_2(1120\tau_2^3 - 5040\tau_2^2 + 6720\tau_2 - 1680) &= \frac{397}{1680}, \\
\frac{1}{3}\omega_1\tau_1^3 + \frac{1}{315}\omega_2(-315\tau_2^3 + 1260\tau_2^2 - 1260\tau_2 + 420) &= \frac{151}{315}
\end{aligned} \tag{26}$$

with the unknowns τ_1 , τ_2 , ω_1 , and ω_2 . Solving it numerically gives

$$\begin{aligned}
\tau_1 &= 0.72289886179270511319, & \omega_1 &= 0.55950733567808927174, \\
\tau_2 &= 1.58789880583487289415, & \omega_2 &= 0.44404926643219107283.
\end{aligned} \tag{27}$$

Let us point out that there are several difficulties with (26). First of all, the system is built under the assumption that $\tau_1 \in [0, 1]$ and $\tau_2 \in [1, 2]$ which in general is not known. In this simple configuration with only four elements, one could, eventually, follow the argumentation of [4] and prove that there has to be a quadrature node in each element. For higher degrees and/or non-uniform knots, however, such an approach is not straightforward. The other difficulty is to have a good initial guess from which the numerical solver converges to the root.

Example 1. Boundary system. For weight functions that are affected by the boundary, a special treatment is needed. Fig. 4 shows the case when B'_2 is the weight. Four basis functions have non-zero overlap with B'_2 and therefore Gaussian rule needs two quadrature points. Exactness of the rule when applied to B'_1 and B'_4 implies that $\tau_1 \in [0, 1]$ and $\tau_2 \in [1, 2]$, respectively. The corresponding algebraic system is then

$$\begin{aligned}
\omega_1(1 - \tau_1)^2 &= \frac{7}{60}, \\
\frac{1}{2}\omega_2(2 - \tau_2)^2 + \frac{1}{6}\omega_1(-9\tau_1^2 + 12\tau_1) &= \frac{1}{3}, \\
\frac{1}{2}\omega_1\tau_1^2 + \frac{1}{24}\omega_2(-24\tau_2^2 + 72\tau_2 - 36) &= \frac{5}{24}, \\
\frac{1}{2}\omega_2(1 - \tau_2)^2 &= \frac{1}{120},
\end{aligned} \tag{28}$$

with the unknowns τ_1 , τ_2 , ω_1 , and ω_2 . Solving it numerically, we obtain

$$\begin{aligned}\tau_1 &= 0.48738056234495278228, & \omega_1 &= 0.44397309239946441113, \\ \tau_2 &= 1.27357129855942870521, & \omega_2 &= 0.22269357426720225554.\end{aligned}\quad (29)$$

Remark 5. *As seen in Example 1, each boundary weight would have to be treated separately as it requires its own system, which is less convenient implementation-wise. For boundary elements, one can use standard Gauss rules instead. Observe that the number of degrees of freedom with support on the boundary is negligible for large meshes. Therefore, from now on, we focus on internal elements and derive weighted Gaussian quadrature rules for them.*

3.2. Weighted Gaussian quadrature – Stiffness matrix terms

To compute the stiffness matrix, one needs to integrate products of derivatives, see (14). Observe that derivatives of the spline basis functions change signs in their support. Therefore, to the best of our knowledge, there is no theoretical result that guarantees existence of the quadrature rule with respect to such a weight function [24]. However, we can build a weighted quadrature by using a piece-wise algebraic system.

The derivatives of the basis functions of degree p still span $p+1$ elements, and there are, same to the mass matrix case, $2p+1$ basis functions that interact with B'_j . However, this spline space is only $2p$ -dimensional, that is, the derivatives that have non-zero support on the support of B'_j are linearly dependent. Having only p quadrature points is not possible due to the symmetry and the fact that the first $p+1$ derivatives are linearly independent on the support of B'_j , see Fig. 5. Therefore, to obtain our sought weighted Gaussian rule, we build the exactness constraints as

$$\int_{\xi_j}^{\xi_j+p+1} B'_i(x) B'_j(x) dx = \sum_{k=1}^{p+1} \omega_k^j B'_i(\tau_k^j) \quad \forall i = j-p, \dots, j+p, \quad (30)$$

that contains $2p+2$ unknowns and $2p+1$ constraints, only $2p$ being independent. Since the support of B'_j is $p+1$ elements, we build the system (30) such that there is one node in each element in the case of uniform knots.

3.2.1. C^1 quadratic elements

If the initial space is C^1 quadratic, the derivatives are piece-wise linear. The situation is antisymmetric and therefore it is natural to impose the antisymmetric constraint on the quadrature rule. We impose the second quadrature point to be the middle point $\tau_2 = \frac{3}{2}$, see Fig. 5. The corresponding algebraic system (30) becomes

$$\begin{aligned}\omega_1(1 - \tau_1) &= -\frac{1}{6}, \\ -2\omega_1\tau_1 + \omega_1 - \frac{1}{2}(1 - 2\tau_1) &= -\frac{1}{3}, \\ (\omega_1 + \omega_2)\tau_1 &= 1,\end{aligned}\quad (31)$$

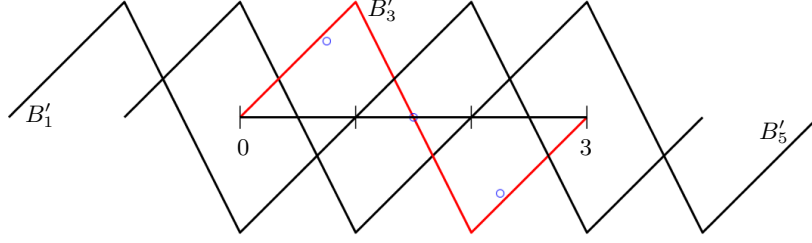


Figure 5: Weighted Gauss quadrature for C^1 quadratic cardinal spline ($p = 2$); stiffness matrix terms. The piece-wise linear derivatives of the quadratic spline basis functions that interact with the weight function B'_3 (red) are shown. The weighted Gaussian quadrature (blue dots, (32)) arises as a solution of the system (31).

with the unknowns τ_1 , ω_1 , and ω_2 . The system has a unique solution

$$\begin{aligned} \tau_1 &= \frac{3}{4}, & \omega_1 &= \frac{2}{3}, \\ \tau_2 &= \frac{3}{2}, & \omega_2 &= 0. \end{aligned} \tag{32}$$

Observe that not imposing antisymmetric constraint on the quadrature rule, see Fig. 5, one obtains an underconstrained (5×6) system whose solution defines a one-parameter family of quadrature rules.

3.2.2. C^2 cubic elements

In the cubic case, one looks for four quadrature points and weights to exactly integrate seven basis functions. This would lead to an underconstrained system, so we again impose the anti-symmetric constraint on the rule (30) and we build a (4×4) system

$$\begin{aligned} \frac{1}{4}\omega_1(1 - \tau_1)^2 &= -\frac{1}{120}, \\ -\frac{1}{2}\omega_2(\tau_2 - 2)^2 + \frac{1}{10}\omega_1(15\tau_1^2 - 20\tau_1) &= -\frac{1}{5}, \\ \omega_1(-\frac{3}{2}\tau_1^2 + \tau_1 + \frac{1}{2}) + \omega_2(2\tau_2^2 - 6\tau_2 + 4) &= -\frac{1}{8}, \\ \frac{1}{2}\omega_1\tau_1^2 + \omega_2(-3\tau_2^2 + 8\tau_2 - 4) &= \frac{2}{3}, \end{aligned} \tag{33}$$

with the unknowns τ_1 , τ_2 , ω_1 , and ω_2 . The system admits one parameter family of solutions, τ_1 being a root of

$$60x^2\omega_1 - 120x\omega_1 + 60\omega_1 - 1 = 0, \tag{34}$$

while ω_1 being the free parameter. Setting, e.g., $\omega_1 = \frac{1}{5}$ gives an admissible solution satisfying the requirements $\tau_1 \in [0, 1]$ and $\tau_2 \in [1, 2]$. Finally, we obtain the rule

$$\begin{aligned} \tau_1 &= 0.71132486540518711775 & \omega_1 &= \frac{1}{5}, \\ \tau_2 &= 1.4446991471727334818 & \omega_2 &= 0.43627696922770090447. \end{aligned} \tag{35}$$

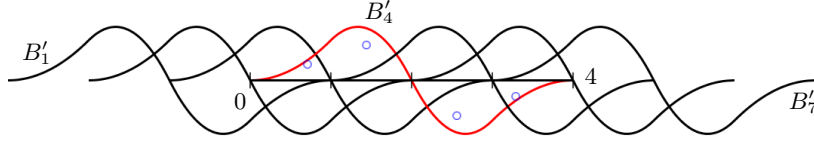


Figure 6: Weighted Gauss quadrature for C^2 cubic splines ($p = 3$); stiffness matrix terms. Blue dots represent the quadrature rule (35) obtained from the system (33).

Remark 6. *Our rules are exact for affine geometric mappings as for them the Jacobians in (14) and (16) are constants. For general (non-affine) maps, our rules compute only approximate values that correspond to a spline approximation of rational functions.*

4. Numerical examples

In this section, we present numerical examples that demonstrate the efficiency and validate the accuracy of the proposed quadrature rules. Our rules exactly integrate the spline spaces (of the parameter domain) associated with the mass and stiffness matrices. Consequently, the rules integrate exactly (up to machine precision) all the matrix entries for constant and affine geometries.

In the following plots, we show the eigenvalue approximation errors that are fundamental for error estimation in many boundary- and initial-value problems [12]. Fig. 7 compares the approximation errors of the standard C^1 quadratic and C^2 cubic isogeometric elements when the mass and stiffness matrices are assembled by the standard Gauss quadrature rule and the weighted Gaussian quadratures proposed in the previous section. In this example, we consider a one-dimensional elliptic eigenvalue problem discretized on a uniform mesh with 1000 elements. The example numerically validates the exactness of our rules as it shows an almost identical error when compared to the standard Gauss integration. The maximum absolute difference in the terms of the mass and stiffness matrices created by both quadratures is of order 10^{-15} .

Fig. 8 shows the convergence of the 2D and 3D eigenvalue problems. We apply the weighted quadrature rules for mass and stiffness matrices for C^1 quadratic ((25) and (32)) and C^2 cubic ((27) and (35)) isogeometric discretizations. The convergence rates obtained numerically are close to the theoretical order of $2p$ for the fully-integrated case [11].

In order to quantitatively compare our approach with [10], we compute the number of floating-point operations and assume that the cost of multiplication and addition operations is the same and is equal to one FLOP. In practice, multiplication can be slightly more costly than addition but we neglect it for now since this phenomenon is architecture-dependent.

We compare the total computational cost of the mass matrix assembly in Fig. 9 for a test 2D problem discretized on a $n \times n$ mesh. The method of [10] requires, due to the reuse of the nodes, asymptotically (for a large number of elements) only two nodes per element in each parameter direction while our

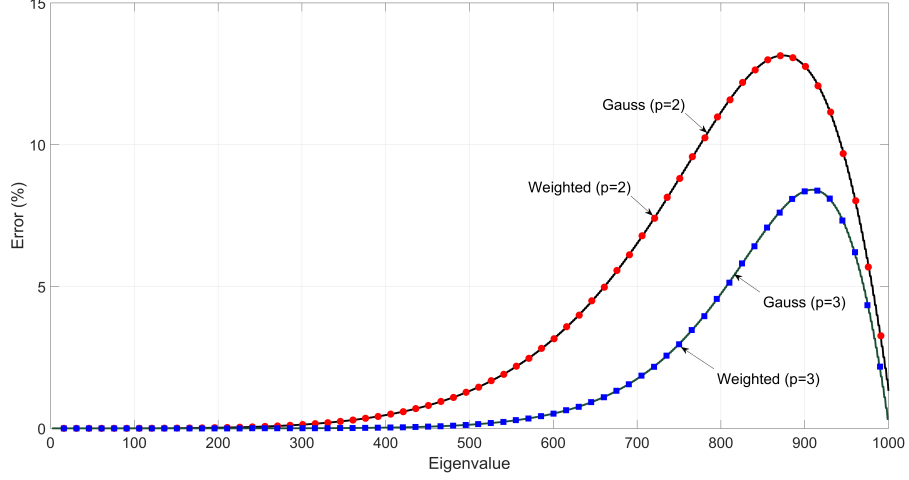


Figure 7: Approximation errors of the quadratic and cubic isogeometric discretizations using the standard Gaussian quadratures (lines) and our weighted quadrature rules (markers). The quadrature rules described in [10] lead to the same results and are not shown here for brevity.

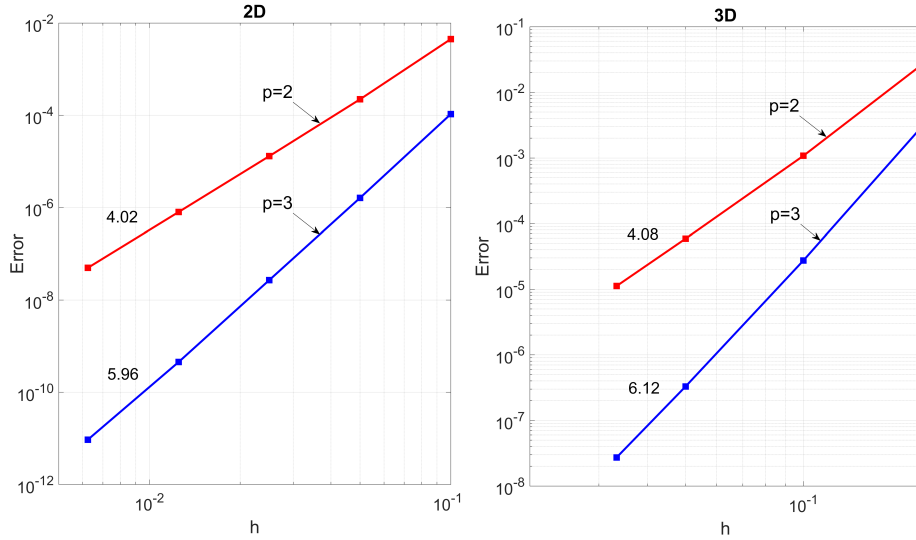


Figure 8: Convergence of the 10-th eigenvalue of the 2D (left) and 3D (right) problems using the quadratic (red line) and cubic (blue line) isogeometric elements. The mass and stiffness matrices were assembled using the weighted quadrature rules introduced in Section 3.

method requires $p + 1$. On the other hand, the FLOPs cost of each integration is less expensive using our approach since the quadrature formula consists of only $p + 1$ summands while [10] needs $2p + 1$. Altogether, we see that FLOPs costs of both methods are similar, with a slight advantage of [10].

Example 2. Consider a 2D mass matrix assembly for $n = 100$, $p = 2$. There

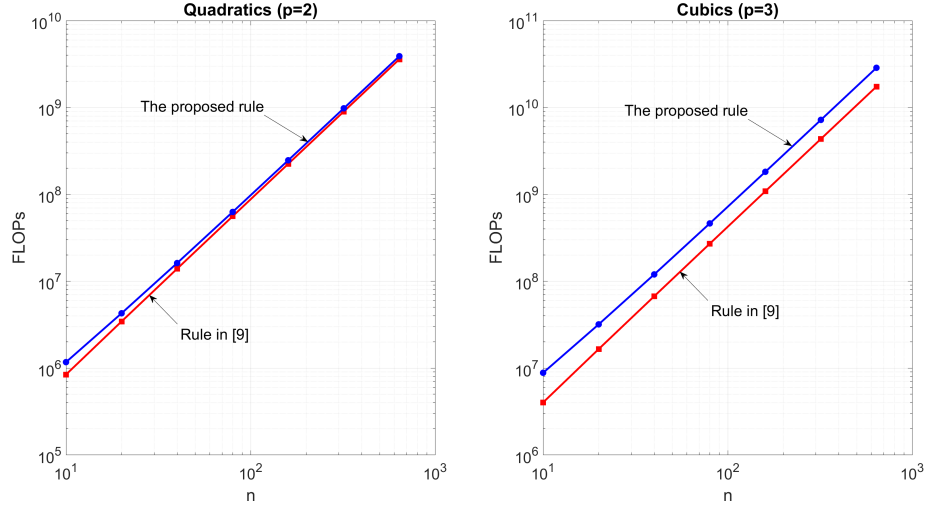


Figure 9: Number of FLOPs; mass matrix. A test 2D problem discretized on a series of $n \times n$ meshes using quadratics (left) and cubic (right) B-spline basis functions. The number of FLOPs needed to assemble the mass matrix using the approach proposed in [10] (red line) and our weighted Gaussian quadrature rule (blue line). We assume periodic boundary conditions for both methods.

are approximately $(2p + 1)n = 500$ non-zero integrals and the evaluation of a quadratic polynomial requires $2A + 2M$ operations, A and M representing addition and multiplication operations, respectively. Then the FLOP estimates read as

$$\begin{aligned} \#FLOP_{wG} &\cong 900(2M + 2A) + 500(3M + 2A) = 3300M + 2800A, \\ \#FLOP_{[10]} &\cong 500(2M + 2A) + 500(5M + 4A) = 3500M + 3000A \end{aligned} \quad (36)$$

since our weighted Gaussian quadrature rule (top row) has only three summands ($(3M + 2A)$ -term) while the approach [10] requires five. Note that (36) is only an estimate (upper bound) as it assumes that the function and the weight have maximum overlap of $p + 1$ elements. Fig. 9 shows the exact number of FLOPs for a 2D test problem assuming $M = A = 1$.

5. Conclusion

We present weighted Gaussian quadrature rules for the mass and stiffness matrix assembly of C^1 quadratic and C^2 cubic spline discretizations over uniform knot vectors. Our rules are the solutions of a set of nonlinear piece-wise polynomial systems. For quadratic elements, the rules factorize to a closed form solution, while for cubics we find solutions numerically. Our weighted quadrature rules require $p + 1$ function evaluations while [10] requires only 2, regardless the degree. For low degrees, we show that this handicap is compensated by the fact that our rules consist of fewer summands and therefore

require fewer floating-point operations (FLOPs). For non-uniform meshes, one can follow the homotopic continuation approach [6] to compute the nodes but such an approach is computationally demanding. We conclude that for uniform meshes, as well as for higher polynomial degrees, the method of [10] offers a more economical solution in terms of FLOPs even though requiring more quadrature points.

As a future work, we aim to follow the analysis conducted in [11, 13, 14, 26] and the fact that the exact integration of the mass matrix is not needed to achieve the optimal convergence rate. Therefore, weighted rules that underintegrate mass terms while using even fewer quadrature points are worth of further investigation.

Acknowledgements

The first author has been partially supported by Spanish State Research Agency (Spanish Ministry of Science, Innovation and Universities): BCAM Severo Ochoa excellence accreditation SEV-2017-0718 and by Ramón y Cajal with reference RYC-2017-22649. This publication was made possible in part by the CSIRO Professorial Chair in Computational Geoscience at Curtin University and the Deep Earth Imaging Enterprise Future Science Platforms of the Commonwealth Scientific Industrial Research Organisation, CSIRO, of Australia. Additional support was provided by the European Union’s Horizon 2020 Research and Innovation Program of the Marie Skłodowska-Curie grant agreement No. 644202.

References

- [1] Adam, C., Hughes, T. J. R., Bouabdallah, S., Zarroug, M., Maitournam, H., 2015. Selective and reduced numerical integrations for NURBS-based isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering* 284, 732–761.
- [2] Aimi, A., Calabro, F., Diligenti, M., Sampoli, M. L., Sangalli, G., Sestini, A., 2018. Efficient assembly based on B-spline tailored quadrature rules for the IgA-SGBEM. *Computer Methods in Applied Mechanics and Engineering* 331, 327–342.
- [3] Auricchio, F., Calabrò, F., Hughes, T. J. R., Reali, A., Sangalli, G., 2012. A simple algorithm for obtaining nearly optimal quadrature rules for NURBS-based isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering* 249-252 (1), 15–27.
- [4] Bartoň, M., Ait-Haddou, R., Calo, V. M., 2017. Gaussian quadrature rules for C^1 quintic splines with uniform knot vectors. *Journal of Computational and Applied Mathematics* 322, 57–70.

- [5] Bartoň, M., Calo, V. M., 2017. Gauss–Galerkin quadrature rules for quadratic and cubic spline spaces and their application to isogeometric analysis. *Computer-Aided Design* 82, 57–67.
- [6] Bartoň, M., Calo, V. M., 2016. Gaussian quadrature for splines via homotopy continuation: rules for C^2 cubic splines. *Journal of Computational and Applied Mathematics* 296, 709–723.
- [7] Bartoň, M., Calo, V. M., 2016. Optimal quadrature rules for odd-degree spline spaces and their application to tensor-product-based isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering* 305, 217–240.
- [8] Bremer, J., Gimbutas, Z., Rokhlin, V., 2010. A nonlinear optimization procedure for generalized Gaussian quadratures. *SIAM Journal on Scientific Computing* 32 (4), 1761–1788.
- [9] Calabrò, F., Manni, C., 2013. The choice of quadrature in NURBS-based isogeometric analysis. In: *Proc. of the 3rd South-East European Conference on Computational Mechanics (SEECCM)*.
- [10] Calabrò, F., Sangalli, G., Tani, M., 2017. Fast formation of isogeometric Galerkin matrices by weighted quadrature. *Computer Methods in Applied Mechanics and Engineering* 316, 606–622.
- [11] Calo, V., Deng, Q., Puzyrev, V., 2019. Dispersion optimized quadratures for isogeometric analysis. *Journal of Computational and Applied Mathematics* 355, 283–300.
- [12] Cottrell, J. A., Hughes, T. J. R., Bazilevs, Y., 2009. *Isogeometric Analysis: Toward Integration of CAD and FEA*. John Wiley & Sons.
- [13] Deng, Q., Bartoň, M., Puzyrev, V., Calo, V. M., 2018. Dispersion-minimizing quadrature rules for c^1 quadratic isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering* 328, 554–564.
- [14] Deng, Q., Calo, V., 2018. Dispersion-minimized mass for isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering* 341, 71–92.
- [15] Fahrenndorf, F., De Lorenzis, L., Gomez, H., 2018. Reduced integration at superconvergent points in isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering* 328, 390–410.
- [16] Gautschi, W., Gori, L., Pitolli, F., 2000. Gauss quadrature for refinable weight functions. *Applied and Computational Harmonic Analysis* 8 (3), 249–257.
- [17] Gomez, H., De Lorenzis, L., 2016. The variational collocation method. *Computer Methods in Applied Mechanics and Engineering* 309, 152–181.

- [18] Hiemstra, R., Calabrò, F., Schillinger, D. and Hughes, T. J. R., 2017. Optimal and reduced quadrature rules for tensor product and hierarchically refined splines in isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering* 316, 966–1004.
- [19] Hughes, T. J. R., Reali, A., Sangalli, G., 2010. Efficient quadrature for NURBS-based isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering* 199 (58), 301 – 313.
- [20] Johannessen, K., 2017. Optimal quadrature for univariate and tensor product splines. *Computer Methods in Applied Mechanics and Engineering* 316, 84–99.
- [21] Kapl, M., Vitrih, V., Jüttler, B., Birner, K., 2015. Isogeometric analysis with geometrically continuous functions on two-patch geometries. *Computers & Mathematics with Applications* 70 (7), 1518–1538.
- [22] Mantzaflaris, A., Jüttler, B., 2012. Exploring matrix generation strategies in isogeometric analysis. In: *International Conference on Mathematical Methods for Curves and Surfaces*. Springer, pp. 364–382.
- [23] Mantzaflaris, A., Jüttler, B., 2015. Integration by interpolation and look-up for Galerkin-based isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering* 284, 373–400.
- [24] Micchelli, C., Pinkus, A., 1977. Moment theory for weak Chebyshev systems with applications to monosplines, quadrature formulae and best one-sided L^1 approximation by spline functions with fixed knots. *SIAM J. Math. Anal.* 8, 206 – 230.
- [25] Oliveira, S. P., Madureira, A. L., Valentin, F., 2009. Weighted quadrature rules for finite element methods. *Journal of Computational and Applied Mathematics* 227 (1), 93–101.
- [26] Puzyrev, V., Deng, Q., Calo, V. M., 2017. Dispersion-optimized quadrature rules for isogeometric analysis: modified inner products, their dispersion properties, and optimally blended schemes. *Computer Methods in Applied Mechanics and Engineering* 320, 421–443.
- [27] Schillinger, D., Hossain, S., Hughes, T., 2014. Reduced Bézier element quadrature rules for quadratic and cubic splines in isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering* 277, 1–45.
- [28] Schoenberg, I. J., 1958. Spline functions, convex curves and mechanical quadrature. *Bulletin of the American Mathematical Society* 64(6), 352–357.
- [29] Xu, G., Li, M., Mourrain, B., Rabczuk, T., Xu, J., Bordas, S. P., 2018. Constructing iga-suitable planar parameterization from complex CAD boundary by domain partition and global/local optimization. *Computer Methods in Applied Mechanics and Engineering* 328, 175–200.